Subject: Add CA certificate authentication to TcpSocket Class
Posted by zouql on Sun, 14 Apr 2024 10:17:18 GMT
View Forum Message <> Reply to Message

Hello Mirek:
I found the HttpRequest and TcpSocket can not support CA certificate function, this may be have the mitm issue. So I add some code to TcpSocket and SSL class for slove this problem.
The steps are as flows:
1. At Core/Inet.h and Core/Socket.cpp

```
String              CAcert;
void        SSLCAcert(const String& cert, bool asn1 = false);

void TcpSocket::SSLCAcert(const String& cert, bool asn1_)
{
   CAcert = cert;
   asn1 = asn1_;
}
```

2. At Core/SSL/SSL.h and Core/SSL/Socket.cpp

```
 class SslContext
{
public:
 SslContext(SSL_CTX *c = NULL);
 ~SslContext()                          { Clear(); }

 bool    IsEmpty() const                { return !ssl_ctx; }

 bool    Set(SSL_CTX *c)                { Clear(); return !!(ssl_ctx = c); }
 bool    Create(SSL_METHOD *meth)       { return Set(SSL_CTX_new(meth)); }
 void    Clear()                        { if(ssl_ctx) { SSL_CTX_free(ssl_ctx); ssl_ctx = NULL; } }
 SSL_CTX *Detach()                      { SSL_CTX *c = ssl_ctx; ssl_ctx = NULL; return c; }

 operator SSL_CTX * () const            { return ssl_ctx; }

 bool    CipherList(const char *list);
 bool    UseCertificate(String certificate, String private_key, bool cert_asn1 = false);
 void    VerifyPeer(bool verify = true, int depth = 2);


 //add by zouql 20240413
 bool    UseCAcert(String CAcert, bool cert_asn1 = false);

private:
 SSL_CTX *ssl_ctx;
};
```

```
bool SslContext::UseCAcert(String CAcert, bool cert_asn1)
{
    ASSERT(ssl_ctx);
    if(IsNull(CAcert))
        return false;
    SslCertificate ca;
    if(!ca.Load(CAcert, cert_asn1))
        return false;

    X509_STORE * castore = SSL_CTX_get_cert_store(ssl_ctx);
    if(castore == NULL)
        return false;

    if(!X509_STORE_add_cert(castore, ca))
        return false;


    return true;
}



bool TcpSocket::SSLImp::Start()
{
 LLOG("SSL Start");

        ...

 //add by zouql 20240413
 if(socket.CAcert.GetCount())
 {
     context.VerifyPeer(true);
     context.UseCAcert(socket.CAcert, socket.asn1);
 }

 return true;
}


dword TcpSocket::SSLImp::Handshake()
{
 int res;
 ERR_clear_error();
 if(socket.mode == ACCEPT)
  res = SSL_accept(ssl);
 else
 if(socket.mode == CONNECT)
```

```
 res = SSL_connect(ssl);
else
 return 0;
if(res <= 0) {
 int code = GetErrorCode(res);
 if(code == SSL_ERROR_WANT_READ)
  return WAIT_READ;
 if(code == SSL_ERROR_WANT_WRITE)
  return WAIT_WRITE;
#ifdef PLATFORM_WIN32
 if(code == SSL_ERROR_SYSCALL && socket.GetErrorCode() == WSAENOTCONN)
#else
 if(code == SSL_ERROR_SYSCALL && socket.GetErrorCode() == ENOTCONN)
#endif
  return WAIT_WRITE;
 SetSSLResError("SSL handshake", res);
 return 0;
}
socket.mode = SSL_CONNECTED;
cert.Set(SSL_get_peer_certificate(ssl));
SSLInfo& f = socket.sslinfo.Create();
f.cipher = SSL_get_cipher(ssl);
if(!cert.IsEmpty()) {
 f.cert_avail = true;
 f.cert_subject = cert.GetSubjectName();
 f.cert_issuer = cert.GetIssuerName();
 f.cert_serial = cert.GetSerialNumber();
 f.cert_notbefore = cert.GetNotBefore();
 f.cert_notafter = cert.GetNotAfter();
 f.cert_version = cert.GetVersion();
 f.cert_verified = SSL_get_verify_result(ssl) == X509_V_OK;
}

if(socket.CAcert.GetCount() > 0)
{
    if(f.cert_verified == false)
    {
        SetSSLError("SSL CA invalid");
    }
}

return 0;
}
```

3. After the change, we can use the HttpRequest like this:
HttpRequest http;
http.Host(m_ip).Port(m_port).Post(strRequest);
http.SSL(true);

String cacrt = "xxxx"; //a string of PEM CA myca.crt
http.SSLCAcert(cacrt);
String strContent = http.Execute();

if the CA cert is not the server.cert's CA ,then report an error.

The CA and Server cert steps:
//1. Gen CA file. myca.key, myca.cert
openssl genrsa -aes256 -out myca.key 2048
openssl rsa -in myca.key -out myca.key
openssl req -new -x509 -days 3650 -key myca.key -out myca.crt -subj "/C=CN/ST=SD/L=YT/O=xx/OU=CA/CN=MYCA/emailAddress=youremail@xxx.com"

//2. gen server.key and server.crt
openssl genrsa -aes256 -out server.key 2048
openssl rsa in server.key -out server.key

openssl req -new -key server.key -out server.csr -subj "/C=CN/ST=SD/L=YT/O=xxx/OU=RD/CN=xxx/emailAddress=youremail@xxx.com"

openssl x509 -req -days 3650 -in server.csr -CA myca.crt -CAkey myca.key -CAcreateserial -out server.crt

File Attachments

1) Socket&SSL.zip, downloaded 151 times